REMARKS

Claims 1-21 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 101 Rejection:

The Examiner rejected claim 11 under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter. The Examiner asserts "claim 11 recites a 'system means'; however, it appears that the system would reasonably be interpreted…as software per se, failing to be tangibly embodied or include any recited hardware as part of the system."

The elements of claim 11 are all expressed as means for performing a specified function. Applicant reminds the Examiner that under 35 U.S.C. § 112, paragraph 6, **by statutory definition**, a means claim must be construed to include the structure that performs the corresponding function. Thus, a means claim **cannot be construed as software *per se***.

Contrary to the Examiner's assertion that "claim 11 recites a 'system means'", claim 11 actually recites *a system, comprising means for creating a plurality of client-side Object Request Brokers (ORBs) for a client of an application server on a client system*. Corresponding structure for the means as recited in claim 11 may be found in the drawings and specification, e.g. in FIG. 1 and FIG. 2 and the description of FIG. 2 beginning on page 9 at line 26, which recites in part:

> Client system 200 may be any of various types of devices, including, but not limited to, a personal computer system, desktop computer, laptop or notebook computer, mainframe computer system, workstation, network computer, or other suitable device… Client system 200 may couple over a network to one or more other devices via one or more wired or wireless network interfaces, such as one or more host machines each hosting one or more application server instances.

Thus, claim 11 must be "construed to cover the corresponding structure" as indicated in 35 U.S.C. § 112, paragraph 6, which according to Applicants' specification includes a physical device.

Therefore, for at least the reasons presented above, the § 101 rejection of claim 11 is improper and removal thereof is respectfully requested.

Section 103(a) Rejection:

The Examiner rejected claims 1, 5, 6, 10-12, 16, 17 and 21 under 35 U.S.C. § 103(a) as being unpatentable over Gigliotti et al. (U.S. Patent 6,393,458) (hereinafter "Gigliotti"). Applicant respectfully traverses this rejection for at least the following reasons.

In regard to claim 1, contrary to the Examiner's assertion, Gigliotti does not teach or suggest *one or more client computer systems configured to implement one or more clients of the application server, wherein each client is configured to: create a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server instances.* The Examiner cites Gigliotti, col. 4 lines 30-31, col. 5 lines 53-56, and FIG. 3 in support of this assertion, and states "wherein each client instance is connected to a server host using ORB." However, Gigliotti, in the cited selections or elsewhere, does not teach or suggest *each client creating a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server instances.* Col. 4, lines 30-34 of Gigliotti reads:

> As used herein, a client object may be an ORB compliant object (such as a CORBA object) and have an associated Graphical User Interface ("GUI") to allow a system user to interact with the software to cause the client object to initiate an event.

Col. 5 lines 53-56 of Gigliotti reads:

> That is, the Event Subscriber need not be located on the same machine as the subscribing object, but only needs to be connected to objects running on a server host through an ORB.

The above citations do not teach or suggest teach or suggest *each client creating a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server instances.* FIG. 3 of Gigliotti shows a plurality of "client objects" coupled to a plurality of "Server hosts", but does not show a plurality of client-side ORBs for each client object, wherein each client-side ORB for each client is coupled to a server-side ORB of a different one of the plurality of application server instances. In contrast, Gigliotti only teaches that each 'client object' may be an ORB. In the description of FIG. 3, beginning at col. 6, line 3, Gigliotti does not teach or suggest anything like *each client creating a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server instances.*

In the Action mailed November 9, 2007, in response to the above arguments, the Examiner asserts "Fig. 3 of Gigliotti clearly indicates multiple client computer systems connected to multiple load balancers with a plurality of client-side ORBs couple to a server-side ORB of a different one of the plurality of application server instances." **The Examiner's interpretation of Gigliotti is clearly incorrect.** Contrary to the Examiner's assertion, Fig. 3 of Gigliotti clearly indicates that each client object is connected to a single instance of a load balancer, and that two client objects may be connected to a single load balancer (e.g., client objects 45 and 46 are both shown as connected to load balancer 52). As noted above, Gigliotti only teaches that each 'client object' may be **an** ORB. Nowhere does Gigliotti teach or suggest what is actually recited in claim 1: *each client is configured to create a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB [created by the client] is coupled to a server-side ORB of a different one of the plurality of application server instances.*

At col. 8, lines 3-24, Gigliotti describes what happens when a client initiates an event via **an** ORB (Emphasis added):

> In an exemplary embodiment as shown in FIG. 4, an event is first initiated by a client 76, possibly by user interaction with a GUI. The event is passed to the Event Publisher 78 **on a first host 70** (via **an** ORB)…Once the event has been initiated by the client, the Event Publisher 78, by means of a load balancing object 80 in communication with Event Publisher 78, adds load balancing information into the event. That is, Event Publisher 78 maintains a list of which machines can run classes that subscribe to an event while load balancer 80 may query the system for particular server hosts such as host 88 in FIG. 4 or hosts 94, 96, and 98 in FIG. 5 that have a low load and Event Publisher 78 can include information relating to that low load host or subscriber in the event in order to direct the event to the lowest load host capable of handling the object. In this way, events can be directed to specific hosts, either for load balancing purposes or because the event has specific processing needs that are met by a specific machine.

From the above and from Fig. 4 and the rest of the description thereof, it is clear that the load balancer described by Gigliotti resides on a host machine that is clearly distinct and different from and located separately from the "client objects" as illustrated in Fig. 3. Gigliotti does not teach or suggest, in reference to Fig. 3, Fig. 4, or elsewhere, anything like *each client creating a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server instances.* In contrast, Gigliotti clearly teaches the notion of load balancers located on host machines which are clearly distinct from clients in Gigliotti's system. Furthermore, Gigliotti only describes a single ORB associated with each client object, and nowhere describes multiple ORBs for each client object, or that clients or client objects create a plurality of client side ORBs. Furthermore, in describing the load balancing system, Gigliotti does not even teach the notion that each load balancer is configured to *create a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server instances.* The Examiner's assertion that "Fig. 3 of Gigliotti clearly indicates multiple client computer systems connected to multiple load balancers with a plurality of client-side ORBs couple to a server-side ORB of a different one of the plurality of application server instances" and that Gigliotti therefore teaches the limitations as recited in claim 1 is clearly **not** supported by the Gigliotti reference.

In the Action mailed November 9, 2007, the Examiner goes on to assert "Fig. 3 clearly indicates a plurality of client-side ORBs for each client object since each client object is coupled with it's own load balancer which in turn uses a client-side ORB to couple to a server-side ORB. In addition, Gigliotti does teach that any communication between two different machines is utilizes ORB…Therefore, one of ordinary skill in the art would have known that each line, in FIG. 3, connecting client 42 to server 44, requires an ORB at both ends." **Whether each line in FIG. 3 "requires an ORB at both ends" is irrelevant**, because nowhere does Gigliotti teach or suggest what is actually recited in claim 1: *each client is configured to create a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB [created by the client] is coupled to a server-side ORB of a different one of the plurality of application server instances.* Furthermore, as indicated above, Gigliotti's load balancers are located on host systems that are clearly distinct from Gigliotti's clients.

In further regard to claim 1, contrary to the Examiner's assertion, Gigliotti does not teach or suggest *wherein each client is configured to select one of the plurality of client-side ORBs according to a load balancing scheme in response to a request to access the application server.* The Examiner cites Gigliotti, col. 6, lines 37-39, in support of this assertion, and states "wherein a load balancer determines a balanced distribution". However, nowhere does Gigliotti teach or suggest *each client selecting one of the plurality of client-side ORBs according to a load balancing scheme in response to a request to access the application server.* Nowhere does Gigliotti teach or suggest each client creating a plurality of client-side ORBs on a client system, and nowhere does Gigliotti teach or suggest each client selecting one of the [created] plurality of client-side ORBs according to a load balancing scheme.

A careful review of the Gigliotti reference reveals that Gigliotti's system is clearly and distinctly different than what is recited in claim 1. In the Summary section (col.2, line 55-col. 3, line 26), Gigliotti describes a "method, system and computer program product for load balancing in a distributed computing environment." Gigliotti discloses

that the system balances the distribution of event messages in a distributed object computing environment. Gigliotti's system includes at least one client publishing an event containing information and a plurality of server classes residing on one or more server hosts, at least one server class subscribing to the event published by the client, and a plurality of load balancers.

Gigliotti's system provides for the registering of a plurality of server classes as subscribers for selected events. For each subscribing server class, the system also provides for the registering of one or more server hosts as capable of running an instance of the class. The client then publishes an event which is received by only one of the plurality of load balancers.

Gigliotti's system then provides for the selection of a server host by one of the load balancers for each subscribing server class registered to subscribe to the event by the load balancer based on load parameters calculated and the registration of hosts capable of running an instance of the subscribing server classes. The selected server host identity for each subscribing server class is then coupled to the event for further publication of the event into the distributed object computing environment by the load balancer. The event is then received by a plurality of event subscribers, each event subscriber reviewing the server host identity information for a server host affiliated with that event subscribers. The event subscribers pass the event on to the identified subscribing class for each class identified for processing on a server host affiliated with the event subscriber. At least one instance of each subscribing class for the published event then performs logical operations in accordance with the event.

Nowhere in the above, or elsewhere, does Gigliotti describe anything like what is recited in claim 1. Nowhere does Gigliotti teach or suggest each client on a client computer system creating a plurality of client-side ORBs, wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server instances, and nowhere does Gigliotti teach or suggest each client selecting one of the created plurality of client-side ORBs according to a load balancing scheme.

In the Action mailed November 9, 2007, in response to the above arguments, the Examiner asserts "The client object coupled to the load balancer of Gigliotti is equivalent to the client as claimed by Applicant since the load balancer components are part of Client 42." **The Examiner's assertion is not supported by the actual teachings of Gigliotti.** Gigliotti states that the "load balancers" are part of a "client side", not of the client itself. See, e.g., col. 6, lines 5-10:

> In the exemplary embodiment, the system 40 is generally divided into a client side 42 and a server side 44. The load balancing system 40 includes a number of calling or client objects 45, 46, 48 and 50 and a number of load balancer objects 52, 54 and 56 which are contained within the client side 42.

From Fig. 4 of Gigliotti and the description thereof as previously cited, it is clear that the load balancer described by Gigliotti resides on a host machine that is clearly distinct and different from and located separately from the "client objects" as illustrated in Fig. 3. Furthermore, even in Fig. 3, it is clear that the "client objects" and the "load balancers" of Gigliotti are different and distinct objects. The fact that Gigliotti logically segregates the "load balancers" into a logical (not physical) "client side" does not make Gigliotti's load balancers part of an actual client of an applications server as is recited in claim 1.

In Gigliotti's system "[t]he client publishes an event which is received by only one of the plurality of load balancers." The separate load balancer on the host would then select a server instance from among a plurality of server instances. In other words, Gigliotti's system using ORBs, a server-side ORB would be selected by a load balancer on the host in response to an event generated by a client. However, even if a "client" in Gigliotti's system has more than one client-side ORB, Gigliotti's system does not select from among a plurality of client-side ORBs to do load balancing. **To the contrary, in Gigliotti's system, a load balancer on the host selects among a plurality of server-side ORBs corresponding to the server instances.** The selected server-side ORB would then communicate with the client-side ORB associated with the client.

In the Action mailed November 9, 2007, in response to the above argument, the Examiner asserts "it is inherent that the corresponding client-side ORB for the selected host be chosen. Furthermore, a client-side load balancer that is coupled to each client that selects from among the servers reads on the claimed invention." Applicants again refer the Examiner to Fig. 4 and col. 8, lines 3-24, cited above, where Gigliotti illustrates and describes what happens when a client initiates an event via an ORB. The description given by Gigliotti of how the load balancing system actually works does not match with the Examiner's interpretation thereof, nor does Gigliotti describe anything like what is recited in claim 1 when viewed as a whole.

The distinctions between Gigliotti and claim 1 of the instant application is clear. **Gigliotti does not teach or suggest each client creating a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server instances, and selecting one of the plurality of client-side ORBs according to a load balancing scheme in response to a request to access the application server**. Instead, Gigliotti teaches a client generating an event, a load balancer receiving the event, and in response to the event, the load balancer selecting from among a plurality of server-side ORBs.

In the Action mailed November 9, 2007, in response to the above argument, the Examiner asserts (emphasis added) "At some point, Gigliotti's invention must create a plurality of client-side ORBs in order to communicate with the servers. Fig. 3 clearly indicates that each client can be coupled to a single load balancer which is then coupled to each of the servers. Gigliotti's invention [the load balancer, not the client object] also must select one of the plurality of client-side ORBs according to a load balancing scheme in response to a request to access the application server." The Examiner simply describes what the Applicants have pointed out: Gigliotti teaches a client generating an event, a load balancer receiving the event, and in response to the event, the load balancer selecting from among a plurality of ORBs. Gigliotti does **not** teach or suggest each client creating a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server

instances, and selecting one of the plurality of client-side ORBs according to a load balancing scheme in response to a request to access the application server, as is recited in claim 1. **And the Examiner's own interpretation of Gigliotti only serves to highlight this distinction.**

Thus, for at least the reasons presented above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 1 apply to claims 6, 11, 12, and 17.

The Office Action rejected claims 2-4, 7-9, 13-15 and 18-20 as being unpatentable over Gigliotti in view of Applicant's Admitted Prior Art (hereinafter "AAPA"). As the rejection of the independent claims have been shown to be unsupported by the cited art, no further comments in regard to these claims is necessary at this time.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-71800/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: January 7, 2008